Representation Space Augmentation for Effective Self-Supervised Learning on Tabular Data

Moonjung Eo, Kyungeun Lee, Hye-Seung Cho, Dongmin Kim, Ye Seul Sim, Woohyung Lim

LG AI Research, Seoul, Republic of Korea
{moonj, kyungeun.lee, hs.cho, dmkim, ysl.sim, w.lim}@lgresearch.ai

Abstract

Tabular data, widely used across industries, remains underexplored in deep learning. Self-supervised learning (SSL) shows promise for pre-training deep neural networks (DNNs) on tabular data, but its potential is hindered by challenges in designing suitable augmentations. Unlike image and text data, where SSL leverages inherent spatial or semantic structures, tabular data lacks such explicit structure. This makes traditional input-level augmentations, like modifying or removing features, less effective due to difficulties in balancing critical information preservation with variability. To address these challenges, we propose RaTab, a novel method that shifts augmentation from input-level to representation-level using matrix factorization, specifically truncated SVD. This approach preserves essential data structures while generating diverse representations by applying dropout at various stages of the representation, thereby significantly enhancing SSL performance for tabular data.

Introduction

Tabular data, organized in rows and columns, is widely used across various fields including finance, healthcare, and marketing. Despite its prevalence, deep learning research has paid less attention to tabular data compared to areas such as computer vision and natural language processing (Chen et al. 2020; He et al. 2020; Grill et al. 2020).

Recently, self-supervised learning (SSL) has emerged as a promising pre-training method for tabular data (Popov, Morozov, and Babenko 2019; Borisov et al. 2022; Yan et al. 2023; Lee et al. 2024). SSL enables models to extract meaningful features and patterns from unlabeled data, which can then be leveraged for various downstream tasks. Many SSL approaches rely on contrastive learning, which employs data augmentation to introduce diverse variations of the input data, enabling models to learn more robust representations (He et al. 2020; Oord, Li, and Vinyals 2018; Onishi and Meguro 2023). This technique encourages the learning of robust and generalizable representations by making models invariant to specific transformations while preserving essential information.

Applying SSL to tabular data is challenging primarily due to the difficulty in designing effective data augmentations. Unlike images or text, which have clear structures that make generating augmentations easier (e.g., cropping or rotating for images, manipulating semantic structures for text), tabular data lacks inherent positional or spatial relationships (Yoon et al. 2020; Somepalli et al. 2021; Margeloiu et al. 2024). In addition, the heterogeneous mix of data types, arbitrarily ordered columns, and complex feature correlations further complicate this problem (Popov, Morozov, and Babenko 2019; Borisov et al. 2022; Yan et al. 2023; Margeloiu et al. 2024; Onishi and Meguro 2023).

Despite the lack of inherent relationships in tabular data, most previous studies have focused on defining augmentations at the *input* level (Somepalli et al. 2021; Darabi et al. 2021; Bahri et al. 2021; Arik and Pfister 2021), such as modifying or randomly removing selected features. While these augmentation methods are straightforward, they might risk corrupting crucial patterns or generating unrealistic data. Consequently, traditional input-level augmentations can undermine the effectiveness of SSL in the tabular domain.

To address these challenges and advance SSL for tabular data, we introduce a novel augmentation method, **RaTab** (**R**epresentation-space **a**ugmentation for **Tab**ular data). RaTab incorporates two key innovations. First, instead of performing augmentation in the raw input space, which often involves complex and difficult to manage relationships between variables, RaTab shifts the augmentation process to the *representation* space. In this space, the model has transformed the raw features into a more structured and organized form, making it easier to manage and augment (Margeloiu et al. 2024). Second, RaTab uses truncated SVD within the representation space to capture the essential structures in the data. This method has been adopted owing to the promising results from unsupervised feature selection techniques, where matrix factorization has been shown to effectively preserve crucial information and identify underlying patterns in the data (Cai, Zhang, and He 2010; Wang et al. 2015; Shi, Du, and Shen 2017). By applying the truncated SVD technique to the encoder's weights, RaTab can effectively identify and preserve important patterns and relationships within the tabular data, which might be difficult to capture in the raw input space. Additionally, to further enhance the diversity of these views, we incorporate dropout techniques (Gao, Yao, and Chen 2021). This combination allows RaTab to fully leverage the representation space, focusing

on crucial data structures while simultaneously increasing the diversity of augmented views, ultimately improving SSL performance on tabular data.

Our main contributions can be summarized as follows. First, we introduce RaTab, a novel representation-space augmentation method designed to effectively leverage the complex relationships inherent in tabular data. Second, we validate the effectiveness of our method through extensive experiments on 13 public tabular datasets, encompassing the classification and regression tasks. Finally, we show that RaTab consistently outperforms both existing deep-learning approaches and tree-based algorithms.

Related Work

SSL has been applied to tabular data primarily through two prominent techniques: autoencoding and contrastive learning. Autoencoding approaches (Yoon et al. 2020; Huang et al. 2020; Majmundar et al. 2022; Lee et al. 2024) focus on reconstructing samples from corrupted versions, aiming to learn robust features adaptable to the heterogeneity of tabular data. These methods employ objectives such as corruption detection or prediction of binning information as pre-text tasks. On the other hand, contrastive learning (Bahri et al. 2021; Ucar, Hajiramezanali, and Edwards 2021; Yoon et al. 2020; Wang and Sun 2022; Darabi et al. 2021; Somepalli et al. 2021) maximizes the similarity between augmented views of the same sample while minimizing similarity to others, using strategies like masking or feature cropping to define positive and negative pairs. VIME (Yoon et al. 2020) utilizes this augmentation to develop pre-training tasks, such as reconstructing the original features and predicting the mask vector from the masked features. In contrast, SCARF (Bahri et al. 2021) employs contrastive learning by comparing the original data with the corrupted version. SubTab (Ucar, Hajiramezanali, and Edwards 2021) and Transtab (Wang and Sun 2022) propose a different data augmentation approach, which involves partitioning the input tabular data into several subsets. Contrary to approaches that augment views in input space, our approach augments in latent space to address the heterogeneity and lack of clear structure in tabular datasets. Contrastive Mix-up (Darabi et al. 2021) takes a different approach, employing manifold mix-up to generate two distinct data views for contrastive learning purposes. Similarly, SAINT (Somepalli et al. 2021) utilizes cut mix and manifold mix-up to create diverse data views, which are then used for contrastive learning. In this study, we define positive pairs using top singular vectors of the representation space.

Methods

We propose RaTab, a novel augmentation technique that generates augmented representation for use in contrastive SSL for tabular data. As illustrated in Figure 1, the overall process of our method comprises an encoder, decoder, and projector, which work together to learn informative representations for the tabular data.

As illustrated in Figure 1a, the input data x_i is mapped to the representations z_i through the encoder network f_{en}. To generate a positive view $z_{\text{aug},i}$ of the representation z_i, we apply RaTab augmentation, as shown in Figure 1b. The original representation $z_i = f_{\text{en}}(x_i)$ and its augmented version $z_{\text{aug},i}$ are then passed through the projection head h_{prj} to compute the contrastive loss, which captures the similarities between z_i and $z_{\text{aug},i}$. Simultaneously, z_i is also fed into the decoder g_{de} to calculate the reconstruction loss, ensuring reconstructions from the original representations. After the SSL phase, we retain only the encoder f_{en} for fine-tuning. The detailed steps of the whole process are outlined in Algorithm 1.

RaTab: Representation Space Augmentation for Tabular Data

The augmentation process initiates with the extraction of weights W from the final layer of encoder f_{en}. The encoder incorporates dropout, introducing minimal diversity to representations.

The extracted weights are then decomposed using SVD:

$$W = U\Sigma V^T \tag{1}$$

where U and V are orthogonal matrices containing the left and right singular vectors, respectively, and Σ is a diagonal matrix containing the singular values in descending order.

Truncated SVD retains top-k singular values and corresponding vectors:

$$W_k = U_k \Sigma_k V_k^T \tag{2}$$

where U_k, Σ_k, and V_k are truncated forms of the original matrices. The hyperparameter k determines the retained rank, controlling preserved essential information in the latent representation. Experiments utilized P, the percentage of k relative to full rank. The results of these experiments are available in the subsequent analysis section.

Truncated weight matrix W_k updates encoder to $\hat{f}_{\text{en}}$. The updated encoder produces augmented representations:

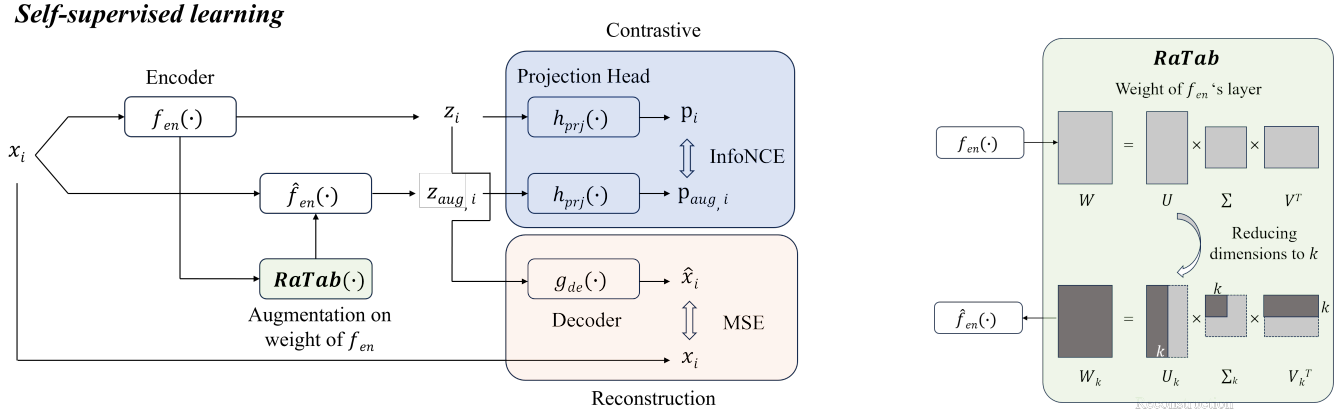
$$\mathbf{z}_{\text{aug}} = \hat{f}_{\text{en}}(\mathbf{x}) \tag{3}$$

where $\mathbf{x}$ represents the input tabular data.

SVD-based augmentation creates meaningful augmentations in latent space, focusing on informative aspects through truncated SVD. Combined with dropout-induced variety, this enhances the learning of robust, generalizable representations. SVD is applied to one layer, typically the last, to manage computational complexity. This approach has shown the best performance in our experiments (see the Analysis section). For large-dimensional networks, randomized SVD (Halko, Martinsson, and Tropp 2011) offers a computationally efficient alternative. We discuss computational considerations in detail in the Ablation section.

Loss Functions We employ a combination of reconstruction and contrastive losses to train the model effectively. The introduction of the reconstruction loss is aimed at modeling intra-example relationships. Given a batch of data X, the encoder f_{en}, and the decoder g_{de}, the reconstruction loss is calculated as follows:

$$\mathcal{L}_{\text{recon}} = \text{MSE}(X, \hat{X}) = \frac{1}{n}\sum_{i=1}^{n} \|x_i - g_{\text{de}}(f_{\text{en}}(x_i))\|_2^2, \tag{4}$$



(a) SSL integrated with RaTab

(b) Augmentation module

Figure 1: An overview of the SSL with our augmentation method, RaTab.

where n is the number of samples in the batch, $\hat{f}_{\text{en}}$ represents the original encoder, and $\hat{X}$ denotes the reconstructed output through g_{de}.

The contrastive loss, implemented using the InfoNCE loss, encourages the model to capture inter-feature differences. Given the original representation $\mathbf{z}$ and the augmented representation $\mathbf{z}_{\text{aug}}$, the contrastive loss is calculated as follows:

$$\begin{aligned}\mathcal{L}_{\text{infoNCE}} &= \text{ContrastiveLoss}(\mathbf{p}, \mathbf{p}_{\text{aug}}) \\ &= -\log \frac{\exp(\text{sim}(\mathbf{p}, \mathbf{p}_{\text{aug}})/\tau)}{\sum_{t=1, t\neq \text{aug}}^{2n} \exp(\text{sim}(\mathbf{p}, \mathbf{p}_t)/\tau)},\end{aligned} \tag{5}$$

where $\mathbf{p} = h_{\text{prj}}(\mathbf{z})$, $\mathbf{p}_{\text{aug}} = h_{\text{prj}}(\mathbf{z}_{\text{aug}})$, h_{prj} is a projection head that maps representations to a lower-dimensional space, and τ is a temperature scaling parameter.

The overall loss function is a weighted combination of the reconstruction and contrastive losses:

$$\mathcal{L}_{\text{overall}} = \lambda \mathcal{L}_{\text{recon}} + (1-\lambda)\mathcal{L}_{\text{InfoNCE}}, \tag{6}$$

where λ is a weighting factor balancing the importance of the two losses. In our experiments, we set $\lambda = 0.5$ to give equal priority to both objectives.

Benefits of RaTab We summarize the benefits of RaTab as follows.

- Representation-level augmentation: Given the lack of inherent relationships in tabular data, applying augmentations in the representation space – which is more structured and organized than raw input – will be beneficial.
- Preserving essential structures with truncated SVD: Effective augmentation requires preserving essential structures in the data. By focusing on the most important directions in the eigenspace through truncated SVD, we can maintain these critical structures.
- Enhancing diversity with dropout: To generate diverse views, it is important to introduce variability in the augmentation process. The stochastic nature of dropout helps achieve this diversity effectively.

Algorithm 1: Self-Supervised learning with RaTab

Input: Dataset D
Parameter: Epochs E, predefined rank k, structure of f_{en}, g_{de}, h_{prj}, loss weight λ
Output: Pre-trained encoder f_{en}

Initialize model parameters of f_{en}, g_{de} and h_{prj}
for $epoch = 1$ to E **do**
  for each batch $X \subset D$ **do**
    $z \leftarrow f_{\text{en}}(X)$
    $W \leftarrow \text{ExtractWeights}(f_{\text{en}}\text{'s last layer})$
    $U, S, V \leftarrow \text{SVD}(W)$
    $W_k \leftarrow U_k \cdot \text{diag}(\text{SelectTopK}(S, k)) \cdot V_k^T$
    Update f_{en}'s last layer weights to W_k
    $z_{\text{aug}} \leftarrow \hat{f}_{\text{en}}(X)$
    $\hat{X} \leftarrow g_{\text{de}}(z)$
    $\mathcal{L}_{\text{recon}} \leftarrow \text{MSE}(X, \hat{X})$
    $p, p_{\text{aug}} \leftarrow h_{\text{prj}}(z), h_{\text{prj}}(z_{\text{aug}})$
    $\mathcal{L}_{\text{InfoNCE}} \leftarrow \text{ContrastiveLoss}(p, p_{\text{aug}})$
    $\mathcal{L}_{\text{total}} \leftarrow \lambda \mathcal{L}_{\text{recon}} + (1-\lambda)\mathcal{L}_{\text{InfoNCE}}$
    Update parameters of f_{en}, g_{de}, and h_{prj} by minimizing $\mathcal{L}_{\text{total}}$
  end for
end for

- Hybrid loss functions: We optimize both contrastive and reconstruction losses during SSL to capture inter-feature and intra-sample dependencies.
- Compatibility with any choice of encoder architecture: RaTab is applied only to the last layer of the encoder network, simplifying integration and ensuring compatibility with any encoder architecture.

Experiments

In this section, we present extensive experimental results to demonstrate the efficacy of RaTab. We compare RaTab with a wide range of SoTA models, including various gradient-boosted decision trees (GBDTs) and DNNs. All experiments were performed on a single NVIDIA GeForce RTX 3090.

Experimental Setup

Datasets We evaluate RaTab on 13 diverse datasets from OpenML (Vanschoren et al. 2014)[1], including binary classification, multiclass classification, and regression tasks. The datasets range from 1,000 to 98,050 samples and 5 to 93 features, covering both categorical and continuous variables. This variety enables comprehensive assessment across data distributions and complexities. Details of these datasets are in Table 5 (Appendix).

Architecture of Neural Networks For the encoder network f_{en}, we experiment with three types of deep networks: MLPs (Multi-Layer Perceptrons), FT-Transformer (Gorishniy et al. 2021) and T2G-Former (Yan et al. 2023). To optimize the depth and width of f_{en} for MLPs, we determine the best configuration based on validation performance in a supervised setup—training only the encoder with a linear head under a supervised loss, which preserves the unsupervised nature of our framework. For the FT-Transformer and T2G-Former, we adhere to the default configurations as specified in their respective original publications. Regarding the decoder g_{de}, we consistently use an MLP architecture identical to the MLP-type encoder network across all experiments. For the projector h_{prj}, we employ a single-layer MLP. Each dataset is trained using consistent architecture and optimization setups. Detailed information on the architectures is provided in Appendix.

Hyper-parameters For each network and dataset, we determine the rank k by selecting the p percentage of the full rank. The value of p is chosen from a set of predefined percentages: $\{50\%, 60\%, 70\%\}$. In other words, we explore using 50%, 60%, or 70% of the full rank of each network to find the optimal value of k. We then determine the optimal configuration based on validation performance for each downstream task. The encoder's dropout rate was fixed at 0.1 for all experiments.

Fine-tuning After self-supervised pre-training, both the decoder g_{de} and projector h_{prj} are removed, and we refine the model using labeled datasets for improved task-specific performance. Fine-tuning unfreezes the encoder for further training alongside a new classifier, allowing direct adjustment of the encoder's weights to better adapt to specific tasks. We used a learning rate of 0.001 for finetuning across all experiments. We report the average results with 10 times finetuning with different random seeds. Other hyperparameters are summarized in Section B in Appendix.

Supervised models - Baseline We conducted a comprehensive evaluation of RaTab applied models across two main baselines. First, we compared the performance of RaTab against traditional GBDT models such as XGBoost (Chen and Guestrin 2016), CatBoost (Prokhorenkova et al. 2018), and LightGBM (Ke et al. 2017). Second, we compared RaTab applied model against several leading DNNs, including MLP, FT-Transformer (Gorishniy et al. 2021), ResNet, NODE (Popov, Morozov, and Babenko 2019), TabNet (Arik and Pfister 2021), DCNv2 (Wang et al. 2021), SAINT (Somepalli et al. 2021), SCARF (Bahri et al. 2021), and T2G-former (Yan et al. 2023). To ensure a fair comparison and avoid the need for extensive hyperparameter tuning, we referenced the reported performance from the original papers. If not provided, we trained the models using default settings as described in their respective papers or left the entries blank in the comparison table.

Augmentation methods We compared our method with four different augmentation techniques: masking, shuffling, and Cut-Mix at the input level, and Contrastive-mix-up in the latent space (Yoon et al. 2020; Ucar, Hajiramezanali, and Edwards 2021; Majmundar et al. 2022; Verma et al. 2019; Darabi et al. 2021). Given a sample $x_i \in \mathbb{R}^d$ in dataset $\mathcal{D}$ with d features and a batch size N, a vector m_i of the same size as x_i is randomly generated. Each element in m_i is independently sampled from a Bernoulli distribution with probability p_m. Depending on how the values are replaced, the method can be classified as either Masking or Shuffling:

Masking:

$$\tilde{x}_{i,k} = m_{i,k} \cdot x_{i,k} + (1 - m_{i,k}) \cdot \mu_k$$

Here, μ_k represents the average of each feature k in the training dataset. In this method, the corrupted feature value is replaced with a pre-determined constant value, which is the average of the respective feature in the training dataset.

Shuffling:

$$\tilde{x}_{i,k} = m_{i,k} \cdot x_{i,k} + (1 - m_{i,k}) \cdot x_{j,k}, \quad j \in \{1, 2, \ldots, N\} \backslash \{i\}$$

Here, j represents the index of another randomly selected sample from the same batch. In this method, the corrupted feature value is replaced with the corresponding feature value from another sample in the same batch.

CutMix:

$$\hat{x} = M \odot x_A + (1 - M) \odot x_B$$

Here, M is a binary mask generated to determine the regions to be mixed from each input sample x_A and x_B. The mixed sample $\hat{x}$ is directly used for training the model.

Contrastive-mixup:

$$\hat{z} = \lambda z_A + (1 - \lambda) z_B$$

Here, z_A and z_B represent the latent representations of two input samples, and λ is drawn from a Beta distribution. The mixed latent representation $\hat{z}$ is then used for training the model.

Comparison with Other Augmentation Techniques

We conducted a comprehensive comparison of our proposed RaTab augmentation strategy with popular input-space and latent-space augmentation techniques. We used the same MLP backbone for all augmentation methods. As shown in Table 1, RaTab demonstrated superior performance across all tasks and datasets compared to other methods. Among the other techniques, contrastive mix-up, operating in latent space, generally outperformed input-space augmentations, indicating the potential benefits of latent-space augmentation. Building on this latent-space advantage, RaTab's consistent outperformance across varied tasks and datasets suggests the effectiveness of its specific approach: rank-based augmentation in latent space.

[1] See https://www.openml.org for more information.

Location of Augmentation	Method	Binary						Multiclass			Regression			
		CH ↑	AD ↑	BM ↑	CS ↑	HI ↑	PO ↑	GE ↑	HE ↑	OT ↑	CPU ↓	DIA ↓	FI ↓	HO ↓
Input	Masking	0.846	0.843	0.895	0.730	0.699	0.878	0.622	0.381	0.784	2.908	704.664	8734.886	5.047
	Shuffling	0.847	0.759	0.892	0.710	0.706	0.893	0.659	0.384	0.796	2.991	764.027	8909.607	3.580
	CutMix	0.845	0.852	0.897	0.730	0.715	0.889	0.629	0.377	0.810	3.063	635.655	8802.745	2.871
Latent	Contrastive-mixup	0.853	0.851	0.897	0.700	0.715	0.902	0.668	0.368	0.807	2.909	570.743	9518.772	2.929
	RaTab	**0.881**	**0.865**	**0.907**	**0.820**	**0.722**	**0.924**	**0.694**	**0.388**	**0.813**	**2.772**	**563.975**	**8451.140**	**2.763**

Table 1: Performance comparison of augmentation techniques. The evaluation metrics for classification and regression are accuracy (↑) and mean squared error (↓), respectively. **Bold** values indicate the best-performances and underlined values represent the second-best performances.

Method	Binary						Multiclass			Regression			
	CH ↑	AD ↑	BM ↑	CS ↑	HI ↑	PO ↑	GE ↑	HE ↑	OT ↑	CPU ↓	DIA ↓	FI ↓	HO ↓
GBDT models													
XGBoost	0.858	0.871	0.813	0.730	0.724	0.876	0.684	0.368	0.808	3.201	856.366	11131.716	3.169
CatBoost	0.855	0.869	0.815	0.760	0.737	0.904	0.729	0.381	0.819	2.607	849.638	9626.546	4.352
LightGBM	0.869	0.868	0.816	0.710	0.731	0.896	0.706	0.368	0.823	2.811	843.905	10622.170	4.519
DNN models													
MLP	0.858	0.849	0.792	0.680	0.720	0.906	0.586	0.375	0.804	2.925	868.154	10754.350	3.173
FT-Transformer	0.867	0.859	0.805	0.730	0.731	0.909	0.690	0.391	0.803	2.791	837.665	9567.579	3.124
ResNet	0.858	0.854	0.802	0.720	0.731	0.880	0.681	0.379	0.806	2.869	853.437	10270.991	4.733
NODE	0.859	0.858	-	-	0.725	-	0.539	0.353	0.803	-	-	-	3.216
TabNet	0.850	0.850	-	-	0.720	-	0.600	0.379	0.791	-	-	-	3.252
DCNv2	0.857	0.853	-	-	0.722	-	0.557	0.385	0.802	-	-	-	3.172
T2G-former	0.863	0.862	0.798	0.730	0.734	0.887	0.656	0.391	0.819	2.878	851.951	10121.356	3.138
SAINT	0.885	0.863	0.807	0.737	0.728	0.866	0.694	0.392	0.825	3.069	794.738	19366.582	5.652
SCARF	0.858	0.842	0.878	0.720	0.653	0.830	0.485	0.197	0.604	3.266	1091.008	9096.269	4.126
RaTab (on MLP)	0.881	0.865	0.907	**0.820**	0.722	**0.924**	0.694	0.384	0.813	2.772	527.028	8451.140	**2.763**
RaTab (on FT-Transformer)	0.880	0.868	0.911	0.790	0.737	0.898	0.676	0.392	0.805	2.588	**498.598**	8715.532	2.934
RaTab (on T2G-former)	**0.888**	**0.873**	**0.912**	0.790	**0.738**	0.907	**0.753**	**0.397**	**0.826**	**2.489**	511.783	**7833.063**	2.856

Table 2: Comparison with GBDTs and DNNs. The evaluation metrics for classification and regression are accuracy (↑) and mean squared error (↓), respectively. The best results are highlighted in **bold** and underlined values represent the second-best performances.

Comparison with SoTA GBDTs and DNNs

Based on Table 2, we can observe several key characteristics when evaluating RaTab against SoTA GBDTs and DNNs. Our method consistently outperformed GBDTs across all datasets, notably achieving significantly higher performance on BM, CS, and DIA datasets. RaTab also demonstrated superior performance compared to SoTA DNNs across all datasets. Furthermore, our experiments with different model backbones showed competitive results even with simple MLP structures, while RaTab frequently achieved top performance when paired with the T2G-former backbone, suggesting a strong synergy between our augmentation technique and this architecture. A more detailed comparison with additional SoTA methods is provided in the Appendix for further reference.

Analyses and Ablation Studies

In this section, we conduct a series of ablation studies and in-depth analyses to uncover the key characteristics of RaTab and elucidate the reasons behind its impressive performance. In addition, we explore of alternatives to reduce the SVD computational load along with their experimental results.

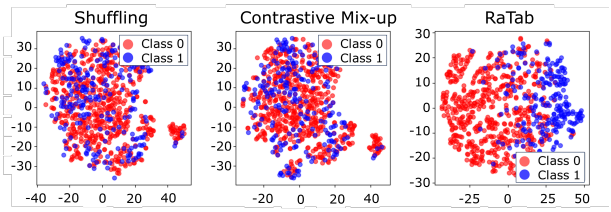


Figure 2: t-SNE visualizations of encoder representations for three different augmentations: Shuffling, Contrastive Mix-up, and RaTab, respectively.

Embedding Analysis

RaTab preserves crucial data structures while enhancing diversity through truncated SVD with dropout, aiming for distinct class separation and diverse intra-class representations. In order to validate these characteristics, we analyzed encoder embeddings through t-SNE visualization and alignment-uniformity metrics. The t-SNE results in Figure 2 demonstrate that RaTab achieves more distinct class differentiation in comparison with input augmentation (Shuffling) and representation augmentation (Contrastive Mix-up). We employed alignment-uniformity metrics proposed by Wang

Figure 3: Alignment-uniformity scores plot for various SSL augmentation methods. The gradient intensity increases with the number of epochs, with darker markers representing later epochs. RaTab shows a consistent decrease in alignment score while maintaining competitive uniformity score. Other methods exhibit higher alignment scores or significant fluctuations. Lower scores for both metrics indicate better performance.

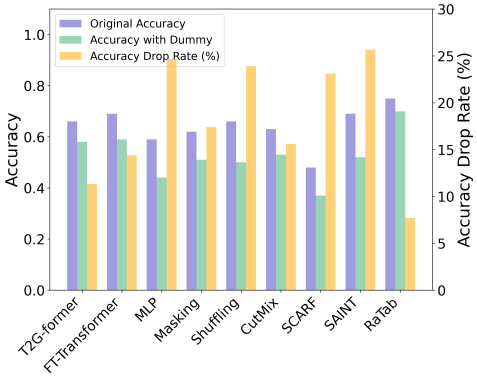


Figure 4: Test accuracy changes when adding uninformative features. Added features are sampled from standard Gaussians uncorrelated with the target and with other features. This experiment was performed on GE data.

and Isola (2020), where lower alignment scores indicate better preservation of critical information and lower uniformity scores reflect increased diversity. Our experimental results in Figure 3 indicate that RaTab consistently reduces alignment scores while maintaining competitive uniformity across training epochs. In contrast to alternative methods that exhibit higher alignment scores or significant fluctuations, RaTab effectively maintains the balance between inter-class distinctiveness and intra-class diversity while preserving embedding space structure. Additional analyses on extended datasets are provided in the Appendix E & F.

Robustness to uninformative features

This experiment is designed to assess our model's ability to remain robust against independent or task-irrelevant information while preserving critical features, a key focus of our methodology's design. By introducing a large number

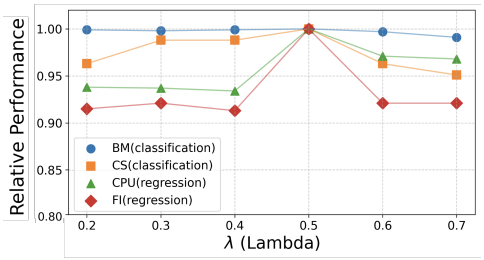


Figure 5: Relative performance across different λ values.

of uninformative features, we aim to simulate the complexity and noise often present in real-world data scenarios. The experiment was conducted by adding dummy features, independent of both labels and original features, in a quantity double that of the original features following the method proposed by Grinsztajn, Oyallon, and Varoquaux (2022). As shown in Figure 4, RaTab demonstrates superior robustness to these uninformative features in tabular datasets. RaTab exhibited a minimal accuracy drop when introducing these dummy features, compared to other models. This robustness extends to scenarios where independent or task-irrelevant information is present, a common occurrence in real-world data. Such characteristics are valuable in practical applications where datasets often contain extraneous variables or unclear feature-target relationships.

Ablation Study on Loss Functions

This section presents the findings from the ablation study that investigated the individual and combined effects of loss functions (see Table 3). For classification tasks, models employing solely contrastive loss generally outperformed those utilizing just reconstruction loss, particularly in datasets such as HI and CS. In contrast, models using only reconstruction loss excelled in regression tasks, especially in datasets like FI and DIA. However, the hybrid model was the standout performer across all evaluations. This model consistently delivered superior performance across all datasets and metrics, highlighting the advantage of a synergistic approach to loss functions. The integration of loss functions provides a more robust and effective solution across different tasks and datasets.

Sensitivity Analysis on Lambda

Throughout this study, we utilized a fixed value for λ as 0.5. In this ablation study, we investigate the impact of the parameter λ in Equation 6 on the performance (with MLP backbone), as shown in Figure 5. For classification tasks, relative performance is the ratio of current to best performance, while for regression tasks, it is the ratio of best to current performance. Our results show that the models exhibit subtle variations in performance across different λ values for both classification and regression tasks. Classification models maintain high performance with slight decreases as λ increases, while regression models experienced a slightly larger performance drop except when $\lambda = 0.5$, at which point both classification and regression models achieved op-

$\mathcal{L}_{\text{recon}}$	$\mathcal{L}_{\text{InfoNCE}}$	Binary						Multiclass			Regression			
		CH ↑	AD ↑	BM ↑	CS ↑	HI ↑	PO ↑	GE ↑	HE ↑	OT ↑	CPU ↓	DIA ↓	FI ↓	HO ↓
✓		0.847	0.853	0.899	0.770	0.676	0.865	0.582	0.371	0.790	2.846	741.129	9128.519	3.225
	✓	0.854	0.856	0.898	0.790	0.718	0.870	0.581	0.371	0.806	2.871	747.455	9177.402	3.250
✓	✓	**0.881**	**0.865**	**0.907**	**0.820**	**0.722**	**0.924**	**0.694**	**0.384**	**0.813**	**2.772**	**527.028**	**8451.140**	**2.763**

Table 3: Results of the ablation study investigating the impact of different loss functions on model performance, using an MLP as the backbone network. We compare the performance of three model variants: contrastive loss only, reconstruction loss only, and the hybrid model combining both losses. The evaluation metrics for classification and regression are accuracy (↑) and mean squared error (↓), respectively. **Bold** values indicate the best-performances, and underlined values represent the second-best performances.

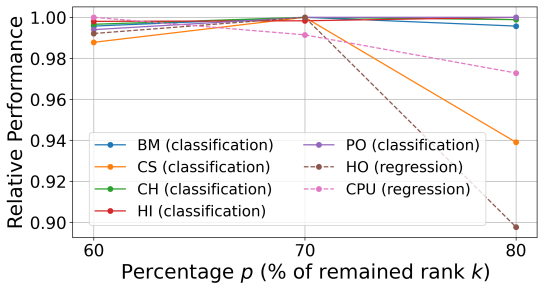


Figure 6: Relative performance across different percentage P values (% of remained rank k).

RaTab applied layer	Classification		Regression	
	PO ↑	GE ↑	DIA ↓	HO ↓
1st layer	0.898	0.667	530.520	2.892
2nd layer	0.909	0.672	527.776	2.800
3rd layer	**0.924**	**0.694**	**527.028**	**2.763**
2nd & 3rd layer	0.909	0.657	528.800	2.800
All layers	0.900	0.657	527.390	2.917

Table 4: Performance comparison of applying RaTab to different layers in the 3-layer MLP model. The evaluation metrics for classification and regression tasks are accuracy (↑) and mean squared error (↓), respectively. The best results for each dataset are highlighted in **bold**.

timal performance. Despite these variations, the overall impact of λ on model performance is relatively moderate, indicating that the models are robust to the tuning of this parameter.

Sensitivity Analysis on k of the RaTab

This analysis presents the results of our experiment on RaTab's performance sensitivity. We conducted tests using percentage p, which represents the proportion of k retained relative to the full rank, to observe how it affects the model's performance across various datasets. Across most datasets, performance fluctuations were minimal as we varied P from 60% to 80%, with optimal performance often occurring at 70%. These findings suggest that using approximately 70% of P in SVD generally provides good model performance. While we observed subtle dataset-specific variations, these were generally small. The model's stability across different P values indicates adaptability to various datasets.

Ablation Study on Layer-wise Application

We investigate the impact of applying RaTab to different layers of the MLP model. The experiments were performed on four datasets (PO, GE, DIA, and HO) using a 3-layer MLP, as it was the deepest architecture used for training these datasets, as shown in Section B in Appendix. The results demonstrate that applying RaTab to the last layer (3rd layer) consistently yields the best performance across all datasets, although the performance differences are generally small. Applying RaTab to multiple layers does not yield further improvements. This can be attributed to the increased information loss when applying SVD to more layers.

Addressing Computational Costs in SVD

Standard SVD has a computational cost of $O(\min(mn^2, nm^2))$, for $m \times n$ matrices, which can be prohibitive for large weights. To address this, we propose using randomized SVD (Halko, Martinsson, and Tropp 2011), which reduces complexity to $O(mn\log(k) + (m+n)k^2)$, where k is the number of retained singular values and vectors. Our experiments show that this approach significantly reduces computation time while maintaining comparable accuracy. Table 8 in the Appendix demonstrates the efficiency gains of randomized SVD, particularly for larger matrices, without compromising performance.

Conclusion

In this paper, we introduce RaTab, a novel representation-level augmentation strategy. RaTab applies truncated SVD to the weight of the representation derived from the data, rather than directly to the input data itself. This approach effectively captures underlying core structures and complex relationships, addressing the challenges posed by the heterogeneous nature of tabular data. It also addresses the challenge in defining what constitutes essential structural information at the input level. Integrating truncated SVD-enhanced representations with contrastive learning, reconstruction techniques, and dropout, RaTab outperforms previous algorithms for tabular domains across various datasets and downstream tasks. Our contributions highlight the potential of RaTab in enhancing the performance of deep learning models on tabular data, paving the way for more robust and effective data augmentation techniques in this domain. Although promising, the effectiveness of RaTab in domains beyond tabular data remains to be validated in future work.

References

Arik, S. Ö.; and Pfister, T. 2021. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, 6679–6687.

Bahri, D.; Jiang, H.; Tay, Y.; and Metzler, D. 2021. Scarf: Self-supervised contrastive learning using random feature corruption. *arXiv preprint arXiv:2106.15147*.

Borisov, V.; Leemann, T.; Seßler, K.; Haug, J.; Pawelczyk, M.; and Kasneci, G. 2022. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.

Cai, D.; Zhang, C.; and He, X. 2010. Unsupervised feature selection for multi-cluster data. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 333–342.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. *Proceedings of the 37th International Conference on Machine Learning*.

Darabi, S.; Fazeli, S.; Pazoki, A.; Sankararaman, S.; and Sarrafzadeh, M. 2021. Contrastive mixup: Self-and semi-supervised learning for tabular domain. *arXiv preprint arXiv:2108.12296*.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 6894–6910.

Gorishniy, Y.; Rubachev, I.; Khrulkov, V.; and Babenko, A. 2021. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34: 18932–18943.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. A.; Guo, Z. D.; Azar, M. G.; et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 21271–21284.

Grinsztajn, L.; Oyallon, E.; and Varoquaux, G. 2022. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in Neural Information Processing Systems*, 35: 507–520.

Halko, N.; Martinsson, P.-G.; and Tropp, J. A. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2): 217–288.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Huang, X.; Khetan, A.; Cvitkovic, M.; and Karnin, Z. 2020. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*.

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Lee, K.; Sim, Y. S.; Cho, H.-S.; Eo, M.; Yoon, S.; Yoon, S.; and Lim, W. 2024. Binning as a Pretext Task: Improving Self-Supervised Learning in Tabular Domains. *arXiv preprint arXiv:2405.07414*.

Majmundar, K.; Goyal, S.; Netrapalli, P.; and Jain, P. 2022. Met: Masked encoding for tabular data. *arXiv preprint arXiv:2206.08564*.

Margeloiu, A.; Bazaga, A.; Simidjievski, N.; Liò, P.; and Jamnik, M. 2024. TabMDA: Tabular Manifold Data Augmentation for Any Classifier using Transformers with In-context Subsetting. *arXiv preprint arXiv:2406.01805*.

Onishi, S.; and Meguro, S. 2023. Rethinking data augmentation for tabular data in deep learning. *arXiv preprint arXiv:2305.10308*.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Popov, S.; Morozov, S.; and Babenko, A. 2019. Neural oblivious decision ensembles for deep learning on tabular data. *arXiv preprint arXiv:1909.06312*.

Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A. V.; and Gulin, A. 2018. CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.

Shi, L.; Du, L.; and Shen, Y.-D. 2017. Unsupervised feature selection via multi-subspace randomization and collaboration. *Knowledge and Information Systems*, 50(1): 207–239.

Somepalli, G.; Goldblum, M.; Schwarzschild, A.; Bruss, C. B.; and Goldstein, T. 2021. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*.

Ucar, T.; Hajiramezanali, E.; and Edwards, L. 2021. Subtab: Subsetting features of tabular data for self-supervised representation learning. *Advances in Neural Information Processing Systems*, 34: 18853–18865.

Vanschoren, J.; Van Rijn, J. N.; Bischl, B.; and Torgo, L. 2014. OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2): 49–60.

Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Lopez-Paz, D.; and Bengio, Y. 2019. Manifold mixup: Better representations by interpolating hidden states. In *International conference on machine learning*, 6438–6447. PMLR.

Wang, R.; Shivanna, R.; Cheng, D.; Jain, S.; Lin, D.; Hong, L.; and Chi, E. 2021. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the web conference 2021*, 1785–1797.

Wang, T.; and Isola, P. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, 9929–9939. PMLR.

Wang, Y.; Li, T.; Huang, X.; Shao, H.; Zhao, H.; and Chen, J. 2015. Unsupervised feature selection via kernel alignment maximization. *IEEE Transactions on Cybernetics*, 45(11): 2581–2592.

Wang, Z.; and Sun, J. 2022. Transtab: Learning transferable tabular transformers across tables. *Advances in Neural Information Processing Systems*, 35: 2902–2915.

Yan, J.; Chen, J.; Wu, Y.; Chen, D. Z.; and Wu, J. 2023. T2g-former: Organizing tabular features into relation graphs promotes heterogeneous feature interaction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 10720–10728.

Yoon, J.; Zhang, Y.; Jordon, J.; and van der Schaar, M. 2020. Vime: Extending the success of self-and semi-supervised learning to tabular domain. *Advances in Neural Information Processing Systems*, 33: 11033–11043.